Proc. 2001 Australian Conference on Robotics and Automation
Sydney, 14-15 November 2001

Model Parameter Identification of Autonomous Vehicles

Tomonari Furukawa and Gamini Dissanayake
Australian Centre for Field Robotics
Rose Street Building, J04
University of Sydney, 2006.
{tomo,dissa}@acfr.usyd.edu.au

Abstract

This paper presents a numerical technique for computing kinematic parameters and calibration constants used for modelling autonomous vehicles. A kinematic model with an accurate parameter set is essential for the proper functioning of the vehicle navigation system. The proposed algorithm does not rely on specific vehicle manoeuvres to compute these parameters. The technique is evaluated using experiments on a Holden-utility vehicle.

1 Introduction

The past few years have seen an increasing interest in the development of autonomous vehicles for outdoor applications [Durrant-Whyte, 1996; Madhavan, et al., 1999; Pilarski, et al., 1999]. A suitable vehicle navigation system that provides the knowledge of vehicle position and trajectory and subsequently controls the vehicle along a desired path is an essential requirement for successful deployment of autonomous vehicles.

The basic navigation system of an autonomous vehicle consists of the dead-reckoning sensors, which are used together with a kinematic model to predict the motion of the vehicle, and the absolute sensors, such as Global Positioning System (GPS) in the case of outdoor vehicles or lasers, which directly measure the vehicle location with respect to its environment. Dead-reckoning sensors accumulate errors due to the integration present in the prediction step [Scheding, et al., 1999]. As absolute information is usually not available at high enough rates to be useful for control purposes, it is important that the dead-reckoning sensors provide accurate information in between such updates.

One of the main sources of error in the location predicted by the dead reckoning is due to the incorrect kinematic parameters and the calibration factors used in the kinematic model. Significant work has been done in identifying the sources of dead-reckoning error and devising benchmarks for measuring such errors in indoor robots accordingly [Borenstein and Feng, 1996]. In most of the previous work, the encoders attached to the wheels and steering are calibrated using only specific manoeuvres such as moving along straight lines and circular paths [Singh and Shin, 1990] and correlating the distance travelled as measured by the encoders and an external measuring device, typically a tape measure, whereas the kinematic parameters are simply based on the original design. These parameters associated with the kinematic model and sensors are thus measured or computed when the vehicle is designed or commissioned. With the notable exception in the on-line estimation of the tyre radius [Durrant-Whyte, 1996], which can change significantly as the load on the vehicle is changed, the parameters are then used as constants within the navigation algorithms. As the characteristics of electro-mechanical systems, however, change with time gradually, the ability to periodically recompute these parameters to check their validity can be of significant benefit.

The goal of the paper is to compute kinematic parameters and calibration constants using data gathered during the normal operation of an autonomous vehicle. This would make it possible to check whether the parameters used in the navigation algorithms are accurate and make any necessary changes without resorting to specific test manoeuvres or modifications to sensor configurations. Such a parameter identification problem is formulated as a minimisation of the difference between the locations (positions and orientations) of the vehicle computed using the kinematic equations and using an absolute position-sensing device. Typically, the information from the dead reckoning sensors are used in the prediction while a GPS receiver is used as an absolute sensor.

Any of the conventional optimisation strategies [Bard, 1974][Dixon, 1974][Press, et al., 1988] can be utilised in the search for the set of parameters by formulating an objective function that minimises a weighted sum of the residuals in position and orientation. However, the solution obtained is obviously dependent on the weighting factors used, and, as the magnitudes of the residues are measured in different units, it is not possible to determine the appropriate weighting factors except through trial and error.

The objective of this paper is to describe a novel strategy for solving optimisation problems that have multiple objectives and demonstrate its effectiveness to solve the

parameter identification problem for autonomous vehicles. In order to solve this class of problems efficiently and in a robust manner, multi-objective continuous evolutionary algorithms (MCEAs), developed using continuous evolutionary algorithms (CEAs) [Furukawa and Dissanayake, 1993; Furukawa and Yagawa, 1997], have been first proposed. The use of the multi-objective optimisation technique allows the parameter set to be found without introducing any additional weighting factors [Fonseca and Fleming, 1995][Furukawa, 2001]. As the multi-objective method finds the solution space rather than a single solution, a technique to select a final solution is further presented.

This paper is organised as follows. Section 2 provides the background material on autonomous vehicles and describes the experimental set up used for obtaining the data. The parameter identification problem for autonomous vehicles is formulated as a multi-objective optimisation problem, and a technique to solve this problem is then presented in section 3. Section 4 presents numerical results to demonstrate the applicability of the proposed technique, and conclusions are summarised in section 5.

2 Autonomous Vehicles

2.1 Experimental Setup

Figure 1 shows a vehicle used as a test bed for research into the navigation of autonomous vehicles. This is a rear wheel driven vehicle that is steered using an Ackerman type steering linkage driving the front wheels.

Figure 1 Vehicle and sensors used.

Four sensors are mounted on the vehicle. An encoder fitted to the rear left wheel gives a measure of the vehicles speed, and a linear variable differential transformer (LVDT) on the steering rack provides a measurement proportional to the steering angle. The encoder and the LVDT are read at a rate of 20 Hz. Carrier Phase Differential GPS unit with a rated accuracy of 0.02 m in position and 0.02 m/s in velocity when at least six satellites are in view is used to measure the absolute position of the vehicle at a sample rate of 4 Hz. An inertial measurement unit comprising of three orthogonal gyroscopes and three accelerometers are also mounted on the vehicle. In the work described in this paper, only one of these gyroscopes is used to measure the angular velocity of the vehicle about a vertical axis. The inertial measurement unit provides information at a sample rate of 125 Hz.

2.2 Vehicle models

The kinematic model of a vehicle moving in the horizontal plane is shown in Figure 2. Location of the vehicle is given by state variables [x , y , ϕ], where x and y are the coordinates of the centre of the rear-axle and ϕ is the orientation of the vehicle body as shown. The inputs that are used to control the vehicle are the velocity at the centre of the rear axle v and the average steering angle γ . The equations of motion for this vehicle at any time instant k is given by:

$$\dot{x}(k) = v(k) \cdot \cos\phi(k) \tag{1a}$$

$$\dot{y}(k) = v(k) \cdot \sin\phi(k) \tag{1b}$$

$$\dot{\phi}(k) = \frac{v(k)}{l} \tan\gamma(k) \tag{1c}$$

where l is the vehicle wheel base.

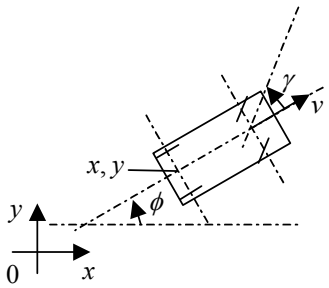


Figure 2 State and control of the vehicle.

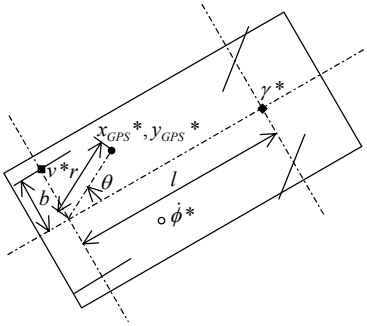


Figure 3 Location and notation of sensors.

2.3 Sensor models

The steering encoder measures the displacement of the steering rack $\gamma_{ENC}*(k)$, which is linearly proportional to the steering angle. The steering angle $\gamma(k)$ can be expressed as

$$\gamma(k) = c_1 \cdot \gamma_{ENC}*(k) + c_2 \tag{2}$$

where c_1 and c_2 are the gain and the offset of the encoder. Meanwhile, the wheel encoder provides the angular position of the left rear wheel of the vehicle, and the difference between successive position measurements can be used to determine the velocity of the left rear wheel. The velocity of the vehicle $v(k)$ is related to the velocity measured by the encoder $v_{ENC}*(k)$ through the following kinematic transformation:

$$v(k) = c_3 \cdot v_{ENC}(k)* + \dot{\phi}(k) \cdot b \tag{3}$$

where c_3 is the gain of the encoder. Note that the substitution of equation (1c) into equation (3) introduces another velocity term $v(k)$. Assembling the velocity terms, resultant velocity $v(k)$ is described as

$$v(k) = \frac{l \cdot c_3 \cdot v_{ENC}*(k)}{l - b \cdot \tan\gamma(k)} \quad (4)$$

The GPS sensor provides the absolute position $[\, x_{GPS}*(k)\,,\, y_{GPS}*(k)\,]$ at which the sensor is mounted. The vehicle location is related to the measurement obtained from the GPS sensor through the equations:

$$x*(k) = x_{GPS}*(k) - r \cdot \cos\left(\phi*(k) + \theta\right) \quad (5a)$$

$$y*(k) = y_{GPS}*(k) + r \cdot \sin\left(\phi*(k) + \theta\right) \quad (5b)$$

where r and θ are the location of the GPS unit, in polar coordinates, with respect to the local coordinate frame on the vehicle as shown in Figure 3.

The rate of change of orientation of the vehicle $\dot{\phi}*(k)$ is related to the reading of the gyroscope $\dot{\phi}_{INS}*(k)$ by

$$\dot{\phi}*(k) = \dot{\phi}_{INS}*(k) + \dot{\phi}_{OFF}* \quad (6)$$

where the initial offset $\dot{\phi}_{OFF}*$ is the average value of the gyroscope measurements obtained *a priori* when the vehicle is stationary:

$$\dot{\phi}_{OFF}* = \frac{\sum_{i=1}^{s} \dot{\phi}_{INS}*(i)}{s} \quad (7)$$

where s is the number of measurements.

2.4 Simulation and Measurement Data

The definitions of the vehicle and sensor models enable us to derive the position and orientation of the vehicle through both the simulation and measurements. Given the state space equations (1), the data to be prepared *a priori* in the simulation are control inputs $[\, v(k)\,,\, \gamma(k)\,]$ with respect to all time (k =1,2,..., k_f) and initial state of the vehicle $[\, x(0)\,,\, y(0)\,,\, \phi(0)\,]$. With the navigation data $[\, \gamma_{ENC}*(k)\,,\, v_{ENC}*(k)\,]$ from the sensors, the control inputs are derived from equations (2) and (3) at any time instant k, and the initial position of the vehicle can also be obtained using equation (5) from measurements by setting $[\, x(0)\,,\, y(0)\,,\, \phi(0)\,] = [\, x*(0)\,,\, y*(0)\,,\, \phi(0)*\,]$. The state of the vehicle at any time instant can now be computed provided that the initial orientation $\phi(0)*$ is specified.

The location of the vehicle $[\, x*(k)\,,\, y*(k)\,,\, \phi*(k)\,]$ can also be computed from measurments at all times. Since the data obtained from the gyroscope is the rate of change of the vehicle orientation $\dot{\phi}*(k)$, the orientation of the vehicle $\phi*(k)$ can be computed through integration, given the initial state $\phi(0)*$. As shown in equation (5), the position is also dependent on $\phi(0)*$ in addition to the measurements from GPS. The location of the vehicle at all times is thus determined by obtaining measurements from the GPS and the gyroscope.

3 Parameter Identification of Autonomous Vehicles

3.1 Problem Formulation

The parameter identification problem can now be characterised as follows:

- Parameters to be identified are $\mathbf{x}^T = [\, c_1\,,\, c_2\,,\, c_3\,,\, l\,,\, b\,,\, \phi(0)*\,,\, r\,,\, \theta\,] \in R^n$.
- The difference between the measured and computed locations of the vehicle must be minimised to identify the parameters.
- For Kalman filter based estimators [Scheding, et al., 1999], the prediction model is only needed to be accurate over each short time period between the receipt of external sensor readings.

To find a solution to this problem, we propose the following multi-objective formulation.

$$\mathbf{f}(\mathbf{x})^T = \left[f_{position}(\mathbf{x}), f_{orientation}(\mathbf{x}) \right] \rightarrow \min_{\mathbf{x}} \quad (8)$$

where, to be accurate over each short time period, objective functions $\mathbf{f}(\mathbf{x}) : R^n \rightarrow R^2$ are given by

$$f_{position}\left(c_1, c_2, c_3, l, b, \phi(0)*, r, \theta\right) \quad (9a)$$
$$= \sum_{i=1}^{n_p} \sum_{j=1}^{k_f'} \left(x(i \cdot k_f' + j) - x*(i \cdot k_f' + j) \right)^2 + \left(y(i \cdot k_f' + j) - y*(i \cdot k_f' + j) \right)^2$$

$$f_{orentation}\left(c_1, c_2, c_3, l, b, \phi(0)*\right) = \sum_{i=1}^{n_p} \sum_{j=1}^{k_f'} \left\| \phi(i \cdot k_f' + j) - \phi*(i \cdot k_f' + j) \right\|^2 \quad (9b)$$

and

$$x(i \cdot k_f') = x*(i \cdot k_f'),\ y(i \cdot k_f') = y*(i \cdot k_f'), \quad (9c)$$

$$\dot{\phi}(i \cdot k_f) = \dot{\phi}*(i \cdot k_f') \quad (9d)$$

where $i = 1,..., n_p$, k_f' is the number of prediction steps in each partition, and n_p is the number of partitions. The total number of iterations is given by $k_f = k_f' n_p$.

While the single-objective optimisation method tries to look for a single solution, the solution of a multi-objective optimisation problem becomes a solution space, any point in which satisfies Pareto-optimality [Coello, 1999]. Since the solution space cannot be obtained analytically, the multi-objective optimisation problem is converted to finding the set of Pareto-optimal solutions, which well describe the solution space iteratively. The next subsection presents MCEAs, which can find such Pareto-optimal solutions efficiently and in a robust manner. A technique to derive a final solution from the Pareto-optimal solutions is also presented.

3.2 Finding the Solution Space

As an efficient and robust multi-objective optimisation method, MCEA proposed here is represented by the following four characteristics; the method:

- Searches with multiple points such that it can find multiple solutions,
- Implements the continuous representation of the points, continuous search formulation and continuous evaluation for efficiency,
- Adopts probabilistic direct search algorithms based on evolutionary computation for robustness,
- Stores all Pareto-optimal solutions historically generated to capture the whole solution space.

A population of individuals, each represented by a continuous vector, is initially (generation t =0) generated at random in the search space $[\, \mathbf{x}_{\min}\,,\, \mathbf{x}_{\max}\,]$, i.e., $P^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_\lambda^t\} \in (R^n)^\lambda$, where λ represents the population size of parental individuals [Furukawa and Dissanayake, 1993]. Each vector thus represents a search point, which corresponds to the phenomenological representation of the individual vector.

The recombination and mutation as evolutionary search

algorithms are defined by the phenomenological evolution accordingly. In the recombination, parental individuals breed offspring individuals by combining part of the information from the parental individuals, thereby creating new points inheriting some information from the old points. The phenomenological recombination is then defined as

$$\begin{cases} \mathbf{x}'_{\alpha} = (1-\mu)\mathbf{x}_{\alpha} + \mu\mathbf{x}_{\beta} \\ \mathbf{x}'_{\beta} = \mu\mathbf{x}_{\alpha} + (1-\mu)\mathbf{x}_{\beta} \end{cases} \tag{10}$$

where parameter μ may be given by the normal distribution with mean 0 and standard deviation σ, i.e., $\mu = N(0,\sigma^2)$, or simply the uniform distribution, $\mu = \text{rand}(\mu_{\min}, \mu_{\max})$. The recombination does not breed offspring individuals considerably different from the parental individuals. An individual is therefore replaced by a random vector with small possibility in the mutation, i.e., $\mathbf{x}'' = \text{rand}(\mathbf{x}_{\min}, \mathbf{x}_{\max})$ [Furukawa and Yagawa, 1997].

Now that a set of new individuals is created, a scalar value (fitness) of each individual is evaluated in the MCEAs for further evolutionary process of selection. First, each individual is ranked in terms of Pareto-optimality with an elimination rule. In the rule, the calculation of objective function at all the points $f_j(\mathbf{x}_i)$ is first conducted, and the Pareto-optimal set is ranked No. 1. The points with rank No. 1 are then eliminated, and Pareto-optimal set in the population is ranked No. 2. All the subsequent ranks are generated stepwise in the same fashion until all the points are ranked. The points in rank No. k, $G(k)$, are defined as

$$G(k) = \left\{ \mathbf{x}_i \mid \text{rank}(\mathbf{x}_i) = k, \forall i \in \{1,...,n\} \right\} \tag{11}$$

for further convenience. At the same time, the normalised fitness of individual $\mathbf{x}_i$ for objective function $f_j(\mathbf{x}_i)$ is calculated as

$$\Phi'_j(\mathbf{x}_i) = \frac{f_{\text{worst}j} - f_j(\mathbf{x}_i)}{f_{\text{worst}j} - f_{\text{best}j}}, \quad 0 \le \Phi'_j(\mathbf{x}_i) \le 1 \tag{12}$$

where

$$f_{\text{best}j} = \min\left\{ f_j(\mathbf{x}_i) \mid \forall i \in \{1,...,n\} \right\} \tag{13a}$$

$$f_{\text{worst}j} = \max\left\{ f_j(\mathbf{x}_i) \mid \forall i \in \{1,...,n\} \right\} \tag{13b}$$

As the fitness of points with the same rank has to be alike, the common normalised fitness is finally assigned:

$$\Phi_j(\mathbf{x}_i) \equiv \Phi_j^{G(k)}(\mathbf{x}_i) = \max\left\{ \Phi'_j(\mathbf{x}_i) \mid \mathbf{x}_i \in G(k) \right\} \tag{14}$$

The fitness of each individual can be conclusively calculated as:

$$\Phi(\mathbf{x}_i) = \sum_{j=1}^{m} \Phi_j(\mathbf{x}_i) \tag{15}$$

Since $\Phi(\mathbf{x}_i) \ge 0$ is satisfied by this equation, the most popular proportional selection can be directly used in the proposed algorithm. In this selection, the reproduction probabilities of individuals are given by their relative fitness:

$$P_s(\mathbf{x}_i) = \frac{\Phi(\mathbf{x}_i)}{\sum_{j=1}^{\lambda} \Phi(\mathbf{x}_j)} \tag{16}$$

So as to capture the configuration of the whole solution space, the resultant Pareto-optimal solutions are stored outside the loop of the evolutionary processes. Points ranked No. 1 in the current population are compared to the Pareto-optimal solutions in the storage, and the Pareto-optimal solutions are updated every time the new points are created.

These evolutionary operations form one generation of the evolutionary process, which corresponds to one iteration in the algorithm, and the iteration is repeated until a given terminal criterion is satisfied.

3.3 Selection of a Final Solution

Finally, there needs to be a technique to select a most appropriate solution from the Pareto-optimal solutions generated by the MCEA. The technique proposed here is the centre-of-gravity method. The final parameter set is the one closest to the centre-of-gravity of the Pareto-optimal solutions although there is no theoretical proof that the centre-of-gravity yields the best parameter set. Given Pareto-optimal solutions $\mathbf{x}_i^*$, $i = 1,..., s$, the centre-of-gravity is given by

$$\mathbf{x}_{cog} = \frac{\sum_{j=1}^{s} \mathbf{x}_j^* \Phi(\mathbf{x}_j^*)}{\sum_{j=1}^{s} \Phi(\mathbf{x}_j^*)} \tag{17}$$

Since all the solutions have the same fitness, i.e., $\Phi(\mathbf{x}_j^*) = \Phi$, the centre-of-gravity is finally expressed as

$$\mathbf{x}_{cog} = \frac{\sum_{j=1}^{s} \mathbf{x}_j^*}{s} \tag{18}$$

From the designer's viewpoint, the parameter set obtained with the centre-of-gravity method can be also considered appropriate since the parameter set, located near the centre of solutions, is robust.

4 Numerical Examples

The proposed technique was applied to the parameter identification of the autonomous vehicle developed by the Australian Centre for Field Robotics. Data from the GPS sensor, the gyroscope, the steering encoder and the velocity encoder were measured while the vehicle was driven in a flat parking lot. The information from these sensors was sub-sampled at 4 Hz to obtain a synchronous sequence of data for the use in the optimisation algorithm. Table 1 lists the various constants used for identification, and the search space of parameters to be identified is listed in Table 2. The search space was chosen to include the original calibration data of each parameter at the centre of the space, and the range was determined based on its reliability.

Table 1 Constants used for identification.

Constant	Value
No. of generations	100
Population	50
Mutation rate	0.10
Time step	0.05
No. of partitions	20

Figure 4 shows the Pareto-optimal solutions in function space after 100 generations. It is seen that the orientation is much smaller than the position in objective function value but that the solutions are well distributed showing a smooth convex-shaped curve in such a different scale. Next, Pareto-optimal solutions in parameter space are depicted in Figure 5. Although the parameter scales are also different

from each other, the solutions in each graph show a characteristic distribution, from upper-left to lower-right for Figure 5(a)-(c) and from lower-left to upper-right for Figure 5(d). The strength of the proposed technique is that we can see the whole solution space and select a solution, which cannot be done by the conventional single-objective optimisation methods.

Table 2 Search space and final solution.

Parameter	l	b	$c_3\ (\times 10^{-4})$	$c_1\ (\times 10^{-4})$
Min	3.10	0.85	4.90	4.50
Max	3.20	1.05	5.00	4.60

c_2	ϕ_0	r	θ
-0.925	1.96	3.65	0.160
-0.900	1.98	3.68	0.190

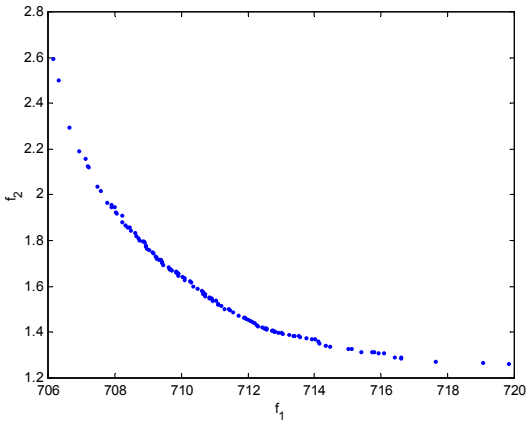
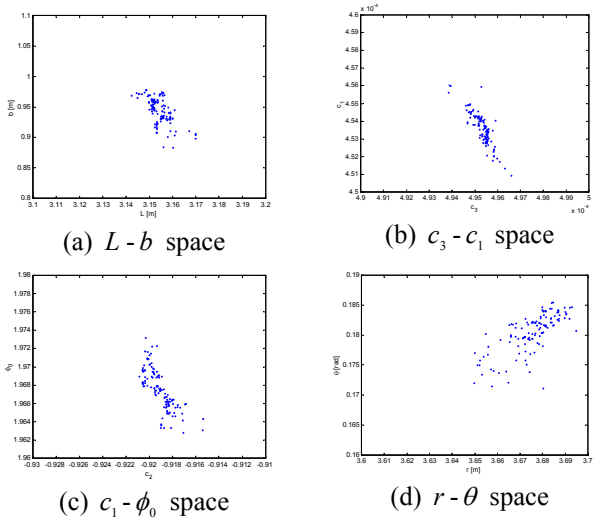


Figure 4 Pareto-optimal solutions in function space.



(a) L - b space (b) c_3 - c_1 space

(c) c_1 - ϕ_0 space (d) r - θ space

Figure 5 Pareto-optimal solutions in parameter space.

Because of the high dimensionality of the parameter space, we chose the final solution by the centre-of-gravity method, and the solution is listed in Table 3. The solutions having the minimum position and orientation errors are also shown in the table for comparison. Since rather monotonic distributions of solutions in parameter space are obtained for this example, the solution chosen by the centre-of-gravity method is well within the solution space.

The simulation result using the parameter set chosen is shown in Figure 6 with the GPS data denoted as 'experiment'. The simulated path shows some accumulated errors, but it is close to the GPS data, indicating clearly that an appropriate parameter set is identified. The differences seen are caused partly by the error accumulation due to the random walk resulting from the integration and partly by the vehicle slip that is not modelled in this example.

To investigate the appropriateness of this solution to the other Pareto-optimal solutions, the above three sets of solutions were used to simulate the current path and the path for the next 100 [s]. The simulation results with the parameters chosen by the centre-of-gravity method are firstly depicted in Figure 7. The figure shows that the results correlate well with GPS data. To investigate the results in more detail, the error values with the three sets of solutions are listed in Table 4. It is seen that the worst solutions in position error and orientation error in both the first and second 100 [s] are the solutions with the minimum orientation error and with the minimum position error, respectively. Particularly, the orientation error by the minimum position error solution and the position error by the minimum orientation error solution, both in the second 100 [s], are significantly large compared to the others. This is certainly caused by the fact that one of the objective functions was ignored. Meanwhile, the solution chosen is not worst in any criterion, and it is even better than the solution with the minimum position error in the position error of the second 100 [s]. This characteristic was seen with different numerical examples.

Table 3 Solutions identified.

Parameter	l	b	$c_3\ (\times 10^{-4})$	$c_1\ (\times 10^{-4})$
Chosen	3.152	0.9441	4.955	4.536
Min ori.	3.160	0.8831	4.946	4.549
Min pos.	3.145	0.9641	4.963	4.513

c_2	ϕ_0	r	θ
-0.9149	1.967	3.675	0.1801
-0.9154	1.963	3.650	0.1770
-0.9203	1.973	3.695	0.1807

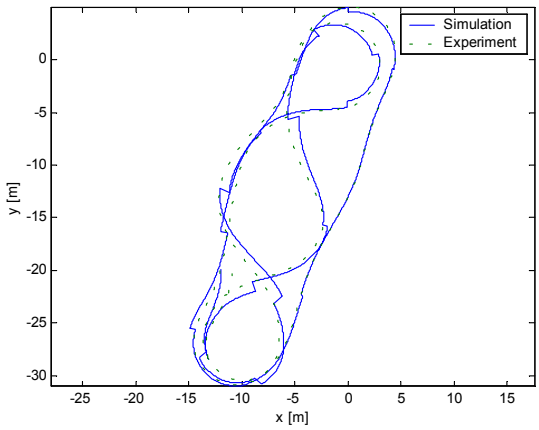
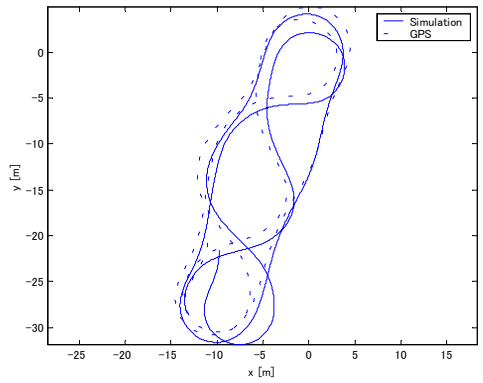
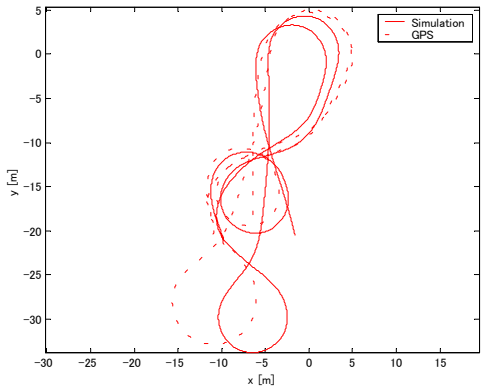


Figure 6 Simulation results with parameters chosen.



(a) 1st 100 [s]



(b) 2nd 100 [s]

Figure 7 Non-partitioned simulation results with parameters chosen.

Table 4 Position and orientation errors.

Solution	Position error		Orientation error	
	1st	2nd	1st	2nd
Chosen	710.6	23,361	1.5656	316.5
Min ori.	706.2	37,992	2.5941	2,674
Min pos.	719.3	253,920	1.2630	8.504

5 Conclusions

A parameter identification technique, consisting of MCEA and the centre-of-gravity method, for autonomous vehicles has been proposed in the paper. The multi-objective formulation in the minimisation of position and orientation errors enables not only the final single solution but also the whole solution space to be derived.

The proposed technique was then applied to parameter identification of an autonomous vehicle, and the solution obtained has been found accurate and robust, minimising both the position and orientation simultaneously. Conclusively, the overall effectiveness of the proposed technique for parameter identification of an autonomous vehicle has been confirmed.

References

[Bard, 1974] Bard, Y. *Nonlinear Parameter Estimation*. Academic Press, New York, 1974.

[Borenstein and Feng, 1996] Borenstein, J. and Feng, L. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics and Automation*. 12(5), October 1996.

[Coello, 1999] Coello, C.A. A Comprehensive Survey of Evolutionary-based Multi-objective Optimization Techniques, *International Journal of Knowledge and Information Systems*. 1999, 1(3), 269-308, 1999.

[Dixon, 1974] Dixon, L.C.W. *Nonlinear Optimisation*. The English Universities Press Limited, 1972.

[Durrant-Whyte, 1996] Durrant-Whyte, H.F. An Autonomous Guided Vehicle for Cargo Handling Applications. *International Journal of Robotics Research*. 15(5), 407-440, 1996.

[Fonseca and Fleming, 1995] Fonseca, C.M. and Fleming, P.J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *International Journal of Evolutionary Computation*. 3(1), 1-16, 1995.

[Furukawa, 2001] Furukawa, T. Parameter Identification with Weightless Regularization. *International Journal for Numerical Methods in Engineering*. 52, 219-238, 2001.

[Furukawa and Yagawa, 1997] Furukawa, T. and Yagawa, G. Inelastic Constitutive Parameter Identification Using an Evolutionary Algorithm with Constitutive Individuals, *International Journal for Numerical Methods in Engineering*. 40, 1071-1090, 1997.

[Furukawa and Dissanayake, 1993] Furukawa, T. and Dissanayake, G. Genetic Algorithms Using Real-valued Strings. *Proceedings of the 71st JSME Annual Meeting*. 930-71, 509-510, 1993.

[Madhavan, et al., 1999] Madhavan, R. Dissanayake, G. and Durrant-Whyte, H.F., Roberts, J.M., Corke P.I. and Cunningham, J. Issues in Autonomous Navigation of Underground Vehicles. *Mineral Resources Engineering*. 8(3), 313-323, 1999.

[Pilanski, et al., 1999] Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K. and Stentz, A. The Demeter System for Automated Harvesting, *Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems*. April, 1999.

[Press, et al., 1988] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. *Numerical Recipes in C*. Cambridge University Press, 1988.

[Scheding, et al., 1999] Scheding, S., Dissanayake, G., Nebot, E. and Durrant-Whyte, H., Autonomous Navigation of an Underground Mining Vehicle. *IEEE Transactions on Robotics and Automation*. 15(1), 85-95, 1999.

[Singh and Shin, 1990] Singh, S. and Shin, D.H. Vehicle and Path Models for Autonomous Navigation. *Vision and Navigation: The CMU Navlab*. C. E. Thorpe, ed., Kluwer Press, April, 1990.